What is claimed is:

1. A computer program product for serializing data structure retrievals and updates, the computer program product embodied on one or more computer-readable media and comprising:

computer-readable program code means for creating two identical tree structures, each representing an initial state for accessing stored data;

computer-readable program code means for performing searches against a first of the two trees;

computer-readable program code means for performing a first update against a second of the two trees, yielding a revised tree;

computer-readable program code means for switching the first tree and the revised tree, such that the first tree becomes the second tree and the revised tree becomes the first tree;

computer-readable program code means for performing, after operation of the computer-readable program code means for switching, a second update against the second tree, yielding a synchronized tree that is structurally identical to the first tree; and

computer-readable program code means for performing subsequent searches against the first tree.

2. The computer program product according to Claim 1, further comprising:

computer-readable program code means for obtaining an exclusive lock prior to operation of the computer-readable program code means for performing the first update; and

computer-readable program code means for releasing the exclusive lock after operation of the computer-readable program code means for performing the second update and the computer-

readable program code means for switching.

3. The computer program product according to Claim 1, wherein atomic transactions are used to maintain proper synchronization between the first tree and the second tree.

4. The computer program product according to Claim 1, wherein the computer-readable program code means for performing the first update further comprises computer-readable program code means for queuing a transaction, and wherein the computer-readable program code means for performing the second update further comprises computer-readable program code means for applying the queued transaction against the second tree that results from operation of the computer-readable program code means for switching.

5. The computer program product according to Claim 1, further comprising computer-readable program code means for performing a subsequent update against the synchronized tree that results from operation of the computer-readable program code means for performing the second update; and wherein operation of the computer-readable program code means for performing the subsequent update causes another operation of the computer-readable program code means for switching.

6. A system for serializing data structure retrievals and updates in a computing environment, comprising:

means for creating two identical tree structures, each representing an initial state for

accessing stored data;

means for performing searches against a first of the two trees;

means for performing a first update against a second of the two trees, yielding a revised tree;

means for switching the first tree and the revised tree, such that the first tree becomes the second tree and the revised tree becomes the first tree;

means for performing, after operation of the means for switching, a second update against the second tree, yielding a synchronized tree that is structurally identical to the first tree; and

means for performing subsequent searches against the first tree.

7. The system according to Claim 6, further comprising:

means for obtaining an exclusive lock prior to operation of the means for performing the first update; and

means for releasing the exclusive lock after operation of the means for performing the second update and the means for switching.

8. The system according to Claim 6, wherein atomic transactions are used to maintain proper synchronization between the first tree and the second tree.

9. The system according to Claim 6, wherein the means for performing the first update further comprises means for queuing a transaction, and wherein the means for performing the second update further comprises means for applying the queued transaction against the second

tree that results from operation of the means for switching.

10. The system according to Claim 6, further comprising means for performing a subsequent update against the synchronized tree that results from operation of the means for performing the second update; and wherein operation of the means for performing the subsequent update causes another operation of the means for switching.

11. A method for serializing data structure retrievals and updates in a computing environment, comprising step of:

creating two identical tree structures, each representing an initial state for accessing stored data;

performing searches against a first of the two trees;

performing a first update against a second of the two trees, yielding a revised tree;

switching the first tree and the revised tree, such that the first tree becomes the second tree and the revised tree becomes the first tree;

performing, after the switching step, a second update against the second tree, yielding a synchronized tree that is structurally identical to the first tree; and

performing subsequent searches against the first tree.

12. The method according to Claim 11, further comprising steps of:

obtaining an exclusive lock prior to performing the first update; and

releasing the exclusive lock after performing the second update and the switching.

13. The method according to Claim 11, wherein atomic transactions are used to maintain proper synchronization between the first tree and the second tree.

14. The method according to Claim 11, wherein the step of performing the first update further comprises queuing a transaction, and wherein the step of performing the second update further comprises applying the queued transaction against the second tree that results from operation of the switching step.

15. The method according to Claim 11, further comprising the step of performing a subsequent update against the synchronized tree that results from performing the second update; and wherein the step of performing the subsequent update causes repeating the switching step.

16. A method of serializing access to data structures in a computing system, comprising steps of:

maintaining two trees, a first of which is used for one or more concurrent searches and a second of which is used for an update operation;

switching the two trees after performing the update operation; and

synchronizing the two trees such that both reflect the update operation.

17. A method of serializing access to data structures in a computing system, comprising steps of:

maintaining two data structures, a first of which is used for one or more concurrent searches and a second of which is used for an update operation;

switching the two data structures after performing the update operation; and

synchronizing the two data structures such that both reflect the update operation.

18. The method of Claim 17, wherein the two data structures are B-trees.